Amendments to the Specification:

Please replace the paragraph beginning at page 3, line 21 with the following rewritten paragraph:

Figure 4a-e is a flow chart of the system of the present invention according to a preferred embodiment;

Please add the following new paragraph after the paragraph ending on line 22 of page 3:

Figure 4b is a continuation of the flow chart of Figure 4a;

Figure 4c is a continuation of the flow chart of Figures 4a and 4b;

Figure 4d is a continuation of the flow chart of Figures 4a, 4b, and 4c;

Figure 4e is a continuation of the flow chart of Figures 4a, 4b, 4c, and 4d;

Please replace the paragraph beginning at page 4, line 25 with the following rewritten paragraph:

Referring to Figure 1, there is shown a client computer 2 accessible by a learner 4. The client computer 2 is any computer capable of running software ~~which is~~ that can access and view content over a computer network, preferably a wide area network such as the Internet 6. Most often, the client computer 2 is a personal computer running a Microsoft Windows or Mac OS operating system and an HTML browser, such as Microsoft Internet Explorer or Netscape Navigator. However, the client computer 2 can take any form, including a handheld device. Other software may also exist on the client computer 2 in order to access multimedia aspects of the learning process, such as streaming audio and video. Also connected to the Internet 6 is a simulation server 8 and. Optionally, a media server 10. The simulation server 8 comprises known HTTP server software, such as Microsoft Internet Information Server running on a Windows NT operating system, as well as software for creating the simulation. The simulation server 8 is responsive to requests from the client computer 2 to view learning content via the Internet 6. The simulation server 8 communicates with a database server 12 in order to provide the learning content. The simulation server 8 is also capable of communicating with multiple client computers 2, 2', 2" for different learners 4, 4', 4". It should be understood that the

physical arrangement of Figure 1 is representative, and the present invention can be implemented in numerous physical arrangements without departing from the scope of the present invention.

Please replace the paragraph beginning at page 6, line 21 with the following rewritten paragraph:

The dynamic user interface generation module 19 further comprises a java servlet 22, a prop producer 24, and a prop ~~26~~ 27. The java servlet 22 receives requests from the client computers 2, 2', 2" and transmits HTML and multimedia data which comprise the user interface of the simulation to the client computers 2, 2', 2" in response to the requests. The java servlet 22 also updates the prop producer 24 in response to actions taken by the learner. The client computer sends a request which takes the form http://servername.com/First Person?Prop=Email&Enabled-true, where "http://"indicates the request is an http request, "servername.com" identifies the simulation server 8, "FirstPerson" identifies the java servlet 22 to be accessed, "?" indicates the beginning of values to be passed to the java servlet 22, "Prop=Email" is a variable and variable value to be passed, "&" indicates the beginning of a new variable and value to be passed, and "Enabled=true" indicates a new variable and value to be passed to the applet.

Please replace the paragraph beginning at page 7, line 5 with the following rewritten paragraph:

The prop producer 24 next communicates with the prop ~~26~~ 27. The prop ~~26~~ 27 then updates the state of the world data structure 20 by manipulating the structure 20 to represent the action taken by the learner 4. For example, if the learner in a simulation decided to open an email, the world data structure would be altered to indicate the email is now open.

Please replace the paragraph beginning at page 7, line 10 with the following rewritten paragraph:

The simulation server 8 also communicates with the cycle director 25. The cycle director 25 controls the process for determining how the simulation is going to react to an action taken by the learner 4. The cycle director 25 first executes the conversation algorithm 26. The combination of the conversation algorithm 26 and the world data structure 20 determine both how to respond to the statement made by the learner 4 and creates a list of statements which the learner can choose from in order to respond to the simulation.

Please replace the paragraph beginning at page 7, line 17 with the following rewritten paragraph:

Conceptually, the present invention uses a tree structure in order to allow the learner to navigate within a simulation. Referring to Figure 2, the tree structure 200 comprises multiple tasks 202, 204. An example of a task is introducing the members of a person's family. Groups of tasks comprise conversations, and tasks are subsections of conversation. An example of a conversation is talking about a person's family. Tasks are ~~be~~ associated with each other in levels, with subtasks of a task called child tasks and higher level tasks called ancestor tasks. Tasks on the same level are sibling tasks. Associated with each task are statements which can be made. Statements may invoke subtasks or transition to sibling tasks. Exploratory conversations have transition statements which allow a learner to move between the tasks. Procedural conversations have ~~a required~~ a required or enforced sequence and, therefore, do not have transition statements. Statements which are associated with a current task are called candidate statements. Tasks may also be specific or directional. Statements presented to the user for selection when the current task is a specific task are the statements associated with the current task and the current task's sibling tasks. Statements presented to the user for selection when the current task is a directional task are the statements associated with the current task; therefore, the only way to move on to an ancestor task when performing a directional task is to complete the task. Finally, tasks may also be described as leaf tasks or non-leaf tasks. Leaf tasks are tasks which have no child tasks. While procedural and exploratory conversations and directional and

A7 cont'd

specific tasks are described, it is to be understood that other types of conversations, tasks and statements can be implemented without departing from the scope of the present invention.